Docket No. AT9-99-483

INSTANT SELECTIVE MULTIPLE SOFT DOCUMENT SHARING BETWEEN MULTIPLE HETEROGENEOUS COMPUTING DEVICES

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention relates generally to the field of computer software and, more particularly, to methods and systems for soft document sharing.

2. Description of Related Art:

Current computing applications are largely single user systems. For example, conventional editing applications allow a single user to open a file and make modifications to the content. If while the file is open by a first user, a second user attempts to open the file, the second user will be prevented from opening or modifying the file. The second user is sometimes permitted to obtain a snapshot copy of the file. The snapshot copy, however, is not updated with any of the subsequent modifications made to the original copy by the first user. Thus, the second user is unable to share in the first user's ideas manifested as file modifications. Moreover, the second user is prevented from modifying the content of the original file and, thus, is prevented from sharing his or her ideas manifested as file modifications. In short, the first and second user are unable to collaboratively edit the file.

Collaboration, as the term is used herein, implies an ability for multiple clients to share ideas. This sharing includes the ability to automatically express one's ideas to the other members without having to have the other members explicitly solicit the ideas. Collaboration also includes the ability for each member to automatically receive any ideas from members who are transmitting ideas. Thus, at a minimum, collaboration implies communication among members that are party to the collaborative effort. This communication/collaboration may follow many models. A "brain-storming" session is an unrestrained model of collaboration. On the other hand, a "round-robin" model, in which each member has a specified turn to express ideas, is a constrained model of collaboration.

To fill this need for collaboration, collaborative software tools have been written that provide a shared document framework that allow documents to be collaboratively accessed by multiple users in a collaborative session. The shared document framework typically provides a mechanism for each collaborator to obtain a replicated copy of the shared document from a network server into the collaborator's local address space, which copy is in synchronism with all other replicated copies of the shared document. Such a system works fine for traditional desktop personal computers connected to a physical network. However, such a model does not work as well for computing devices such as; for example, personal digital assistants that do not have a physical network.

Personal digital assistants (PDAs) and similar computing devices are increasingly used by users to create and maintain schedules of meetings, notes, charts, lists, and other items that the user may need with them at all times. Because of the portable nature of these machines, they provide a convenient way for users to maintain access to these items without being tied to an office. Many of these documents are, despite their portable nature, documents for which collaborative involvement is sometimes needed. For example, during a meeting, many users may need to access and modify a document simultaneously in order to create a presentation. The collaboration of several individuals working on a single document simultaneously may result in a presentation that is markedly superior to one created by a single individual or even by multiple individuals working on the document at separate times. However, currently, PDAs and similar devices do not have the functionality necessary to provide collaborative sharing of documents. Therefore, a method and system of allowing users of PDAs and other portable computing devices that lack a connection to a physical network to collaborate with each other through shared access to soft documents is desirable. Furthermore, a method and system of allowing devices utilizing different operating systems to share documents with each other in a secure environment is also desirable.

SUMMARY OF THE INVENTION

The present invention provides a system and method for providing dynamically shared documents to multiple computing devices. In a preferred embodiment, the system includes a hub and a plurality of computing devices in physical proximity with the hub. Each of the plurality of computing devices communicates with the hub via a wireless connection. The hub acts as a pass-through device receiving and transmitting requests from a requesting computing device to other computing devices and receiving and transmitting answers from the other computing devices to the requesting computing device. Each computing device translates requests and shared documents into a system independent language before transmitting the request of the document to the hub and each computing device translates received documents from the system independent language into a data format preferred by that particular computing device. Thus, documents may be securely shared between heterogeneous computing devices (HCDs), which are computing devices having different operating systems and data formats, without having a physical network connection. Furthermore, the hub is portable, thus allowing the sharing of documents at any location in which several users have gathered for a meeting.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 depicts a pictorial diagram illustrating a document sharing system **100** for portable computing devices in accordance with a preferred embodiment of the present invention;

Figure 2 depicts a block diagram of a hub in which the present invention may be implemented;

Figure 3 depicts a block diagram of a data processing system in which the present invention may be implemented;

Figure 4 depicts a block diagram of an exemplary personal digital assistant in accordance with a preferred embodiment of the present invention;

Figure 5 depicts a flowchart illustrating the processes of the present invention performed on a hub;

Figures 6A and 6B, ~~**Figure 6**~~ depicts a flowchart illustrating a method of sending a request for and receiving a document from another HCD in accordance with the present invention;

Figure 7 depicts a flowchart illustrating a method of processing on a computing device for processing a request for a document in accordance with the present invention;

Figure 8 depicts a flowchart illustrating a method on a hub of re-transmitting requests for documents and the requested documents to computing devices in accordance with the present invention; and

Figure 9 depicts a flowchart illustrating a method on a hub of determining whether other hubs are active in the vicinity of the present hub in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, and in particular with reference to **Figure 1**, a pictorial diagram illustrating a document sharing system **100** for portable computing devices is depicted in accordance with a preferred embodiment of the present invention. Document sharing system **100** includes a hub **102**, laptop computers **104**, **106**, and **108**, and personal digital assistants (PDAs) **110**, **112**, and **114**.

Hub **102** is a portable device that can be carried with a user to any location to facilitate sharing of soft documents between multiple computing devices. Once hub **102** is turned on, it polls the area in which it is located to determine if there is another hub present. If there is another hub present, then hub **102** presents the user with an error indication through, for example, an LED display or through creation of a sound indicating that another hub is present. Once the error indication has been presented to the user, hub **102** then powers down. Alternatively, if hub **102** determines that one or more other hubs are present in the vicinity, hub **102** can collaborate with the other hubs to provide document sharing to the computing devices present in the vicinity.

Once hub **102** has been powered up and determined that there are no other hubs in the vicinity, then hub **102** creates a list of computing devices that are present in the vicinity. As each portable computing device comes into proximity with hub **102**, it will initiate contact with hub **102** to indicate its presence such that hub **102** can add it to the list of computing devices present in the vicinity.

Hub **102**, each of laptop computers **104**, **106**, and **108** and each of PDAs **110**, **112**, and **114** is equipped with a wireless receiver and transmitter for facilitating communications between the computing devices and hub **102**. Preferably, the wireless receiver and transmitter is an infrared receiver and transmitter. However, alternatively, the wireless receiver and transmitter could be any type of wireless receiver and transmitter, such as, for example, a radio frequency (RF) receiver and transmitter. If a wireless receiver and transmitter other than an infrared wireless receiver and transmitter is utilized, the transmission power for the transmitter may need to be limited such that the signals transmitted between the computing devices and hub **102** may not be received and interpreted by devices that are not in the local vicinity of hub **102** to prevent an unauthorized user from intercepting and reading the exchanged documents.

Optionally, as an added security feature in order to prevent unauthorized access to shared documents, data transmission between the computing devices **104**, **106**, **108**, **110**, **112**, and **114** may be encrypted. Methods of encrypting and decrypting data are well known in the art.

When a user of a computing device, such as PDA **110**, that has registered with hub **102** desires to retrieve a document from another computing device, such as laptop computer **108**, that is registered with hub **102**, the requesting computing device sends a request for the document to hub **102**. The request is sent in a commonly understood data format. Hub **102** then retransmits the request to laptop computer **108**. Laptop computer **108** then retrieves the requested document and translates it into the commonly understood file format and sends the document to hub **102**. Hub **102** retransmits the requested document to the requesting computing device, PDA **110**, which then translates the document from the commonly understood data format into the preferred file format of PDA **110**.

Because each computing device translates requests and documents into a commonly understood language before broadcasting the request or document, documents may be shared between heterogeneous computing devices (HCDs) which are computing devices that may be running different operating systems and utilizing different data file formats. For example, laptop computer **108** may be running Microsoft Windows 98 operating system, laptop computer **106** may be running IBM's OS/2 operating system, laptop computer **104** may be running Apple's System 7 operating system, and PDAs **110**, **112**, and **114** may be running 3Com's Palm OS operating system. Yet, because each device includes a translator to translate requests and documents into and out of a commonly understood language, documents created on one machine may be shared with other types of machines.

Users of each registered computing device may control access to documents on their respective machines by designating certain documents as "shared" and allowing others access to only these "shared" documents.

Hub **102** does not store or modify the requests or the documents it receives, but merely acts as a pass-through device that retransmits any requests or documents such that the appropriate party may receive the request or document. By utilizing hub **102**, a computing device is able to communicate with more than one other computing device without having to reorient the device to aim it at the next computing device with which it wishes to communicate.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 1** may vary depending on the implementation. For example, more, fewer, or different types of computing devices may be included without departing from the scope and spirit of the present invention. Furthermore, the depicted example is not meant to imply architectural limitations with respect to the present invention.

With reference to **Figure 2**, a block diagram of a hub in which the present invention may be implemented is illustrated. Hub **200** is an example of a hub, such as hub **102** in **Figure 1**, for facilitating secure wireless sharing of documents between computing devices running different operating systems and utilizing different document formats.

Hub **200** includes a system memory **204**, a processor **202**, a wireless communication interface **206**, and a bus **208**. Optionally, hub **200** may also include a display **212**. Display **212**, if included, allows hub **200** to indicate error and status messages to users.

System memory **204** is used to store a list of computing devices that have registered with hub **200** to share documents with other computing devices. Wireless communication interface **206** is used to facilitate reception and transmission of requests and documents to and from registered computing devices. Hub **200** does not store or alter requests for documents or the documents it receives from a computing device. Hub **200** merely retransmits the request or the document such that the appropriate computing device may receive the request or document.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 2** may vary depending on the implementation and that the depicted example is not meant to imply architectural limitations with respect to the present invention.

With reference now to **Figure 3**, a block diagram of a data processing system in which the present invention may be implemented is illustrated. Data processing system **300** is an example of a laptop computer, such as, for example, any one of laptop computers **104**, **106**, or **108**. Data processing system **300** may be, for example, an IBM Thinkpad notebook computer running a PCI application. Data processing system **300** employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures, such as Micro Channel and ISA, may be used. Processor **302** and main memory **304** are connected to PCI local bus **306** through PCI bridge **308**. PCI bridge **308** may also include an integrated memory controller and cache memory for processor **302**. Additional connections to PCI local bus **306** may be made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter **310**, SCSI host bus adapter **312**, and expansion bus interface **314** are connected to PCI local bus **306** by direct component connection. In contrast, audio adapter **316**, graphics adapter **318**, and audio/video adapter (A/V) **319** are connected to PCI local bus **306** by add-in boards inserted into expansion slots. Expansion bus interface **314** provides a connection for a keyboard and mouse adapter **320**, Infrared (IR) interface **322**, and additional memory **324**. In the depicted example, SCSI host bus adapter **312** provides a connection for hard disk drive **326**, tape drive **328**, CD-ROM drive **330**, and digital video disc read only memory drive (DVD-ROM) **332**. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors.

An operating system runs on processor **302** and is used to coordinate and provide control of various components within data processing system **300** in **Figure 3**. The operating system may be a commercially available operating system, such as OS/2, which is available from International Business Machines Corporation. "OS/2" is a trademark of International Business Machines Corporation. An object oriented programming system, such as Java, may run in conjunction with the operating system, providing calls to the operating system from Java programs or applications executing on data processing system **300**. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on a storage device, such as hard disk drive **326**, and may be loaded into main memory **304** for execution by processor **302**.

IR interface **322** provides a channel for information to be sent from data processing system **300** to an IR transmitter to be transmitted to hub **102** and also provides a channel for information to be received by data processing system **300** from hub **102** via an IR receiver. Communication between data processing system **300** and hub **102** may use, for example, an Infrared Data Association (IrDA) Infrared Communications Protocol (IrCOMM). When data processing system **300** requests a document from another computing device that has established a session with hub **102**, data processing system **300** translates its request into a system independent language or commonly understood file format utilized by hub **102**. The system independent language may be, for example, Java or Extensible Markup Language (XML). Once the document has been translated into a system independent language, data processing system **300** sends the translated request to hub **102** via an IR signal emitted from IR interface **322**. When data processing system **300** receives the requested document from hub **102** via IR interface **322**, the requested document is translated from the system independent language into the format preferred by data processing system **300**. If data processing system **300** receives a request for a document from hub **102**, data processing system **300** retrieves the document and translates the document into the system independent language utilized by hub **102** and sends the translated document to hub **102** via an IR link.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 3** may vary depending on the implementation. For example, other peripheral devices, such as optical disk drives, modems, and the like, may be used in addition to or in place of the hardware depicted in **Figure 3**. The depicted example is not meant to imply architectural limitations with respect to the present invention. For example, the processes of the present invention may be applied to multiprocessor data processing systems.

Turning now to **Figure 4**, a block diagram of an exemplary personal digital assistant **400** is depicted in accordance with a preferred embodiment of the present invention. Personal digital assistant (PDA) **400** may be implemented as any one of PDAs **110**, **112**, or **114** depicted in **Figure 1**. Communication between PDA **400** and hub **102** use, for example, Infrared Data Association (IrDA) Infrared Communications Protocol 1.0 (IrCOMM 1.0). PDA **400** is preferably a palm top computer having a wireless network interface. PDA **400** may be, for example, a Palm VII Palm Pilot from 3Com. PDA **400** has a central processing unit **495** connected to a bus **490**. Central processing unit ("CPU") **495** performs most of the computing and logic functions of PDA **400**. A memory **496** is connected to bus **490**, which stores information to be provided to CPU **495** or otherwise used by PDA **400**. An input/output device **497**, such as a keyboard, is also connected to bus **490** which allows a user to input data for storage in memory **496** or for use by CPU **495**. A display **498** is connected to bus **490**. PDA **400** also has a wireless communication interface **499** for communication with hub **102**. Wireless communication interface **499** may utilize, for example, an infrared or radio frequency transmitter and receiver.

An operating system runs on CPU **495** and is used to coordinate and provide control of various components within PDA **400** in **Figure 4**. The operating system may be a commercially available operating system, such as the Palm OS operating system available from 3Com Corporation. An object oriented programming system, such as Java, may run in conjunction with the operating system, providing calls to the operating system from Java programs or applications executing on data processing system **400**. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on a storage device, such as hard disk drive **494**, and may be loaded into system memory **496** for execution by CPU **495**.

When PDA **400** requests a document from another computing device that has established a session with hub **102**, PDA **400** translates its request into a system independent language or commonly understood file format utilized by hub **102**. The system independent language may be, for example, Java or Extensible Markup Language (XML). Once the document has been translated into a system independent language, PDA **400** sends the translated request to hub **102** via an IR signal emitted from wireless communication interface **499**. When PDA **400** receives the requested document from hub **102** via wireless communication interface **499**, the requested document is translated from the system independent language into the format preferred by PDA **400**. If PDA **400** receives a request for a document from hub **102**, PDA **400** retrieves the document and translates the document into the system independent language utilized by hub **102** and sends the translated document to hub **102** via an IR link.

It should be noted that translation of documents between different data formats is well known in the art.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 4** may vary depending on the implementation. For example, other devices may be used in addition to or in place of the hardware depicted in **Figure 4**. The depicted example is not meant to imply architectural limitations with respect to the present invention.

Referring now to **Figure 5**, a flowchart illustrating the processes of the present invention performed on a hub is depicted. Once the hub is powered up, it performs system checks (step **502**) which include polling the immediate vicinity to determine if any other hubs are present. If not, then it determines if the next event is a poll a heterogeneous computing device (HCD) event (step **504**). If the event is a poll HCD event, then the hub polls present HCDs at a predetermined interval to determine the identities of all HCDs present in the vicinity of the hub (step **506**). The list of present HCDs is updated (step **508**) and the hub waits for the next event (step **503**).

If the event is not a poll HCD event, then the hub determines whether the event is request to retrieve a list of present HCDs (step **510**). If the event is a list of present HCDs, then the hub broadcasts the list of present HCDs to the requesting HCD (step **512**) and then waits for the next event (step **503**).

If the event is not a request for a list of present HCDs, then the hub determines if the event is an HCD requesting a shared document program from another HCD (step **514**). If the event is a request for a shared document, then the hub broadcasts the request to the target HCD (step **516**) and then waits for the next event (step **503**).

If the event is not a request for a shared document, then the hub determines if the event is an HCD sending a shared document to another HCD (step **518**). If the event is an HCD sending a shared document to another HCD, then the hub broadcasts the shared document to the target HCD (step **520**) and then waits for the next event (step **503**).

If the event is not an HCD sending a shared document to another HCD, the hub determines if the event is a power off event (step **522**). If the event is not a power off event, the hub waits for the next event (step **503**). If the event is a power off event, then the hub powers down (step **522**).

Referring now to ~~**Figure 6**~~ Figures 6A and 6B, a flowchart a flowchart illustrating the processes of the present invention performed on a hub. To begin, the HCD receives an event (step **601**). Next, the HCD determines if the event is a request for a list of present HCDs from the hub (step **602**). If the event is a request for a list of present HCDs, then the HCD packs the request (step **604**) and sends the request to the hub (step **606**). The HCD then waits for the next event (step **601**).

If the event is not a request for a list of present HCDs, then the HCD determines if the event is receipt of a list of present HCDs from the hub (step **608**). If the event is a receipt of list of present HCDs from the hub, then the HCD unpacks the request (step **610**), processes the list (step **612**), and waits for the next event step **601**).

If the event is not a receipt of a list of present HCDs from the hub, then the HCD determines if the request is a list of shared documents present on another HCD (step **614**). If the event is a request for a list of shared documents present on another HCD, then the HCD packs the request in a system independent language (step **616**) and sends the request to the hub (step **618**). The HCD then waits for the next event (step **601**).

If the event is not a request for a list of shared documents present no another HCD, then the HCD determines if the event is a list of shared documents on another HCD received from the hub (step **620**). If the event is a list of shared documents present on another HCD received from the hub, then the HCD unpacks the request from the system independent language into a language preferred by the HCD (step **622**) and then processes the list (step **624**). The HCD then waits for the next event (step **601**).

If the event is not a list of shared documents on another HCD received from the hub, then the HCD determines if the event is a request of a shared document from another HCD (step **626**). If the event is a request of a shared document from another HCD, then the HCD packs the request in a system independent language (step **628**) and sends the request to the hub (step **630**). The HCD then waits for the next event.

If the event is not a request for a shared document from another HCD, then the HCD determines if the event is a shared document from another HCD received from the hub (step **632**). If the event is a shared document from another HCD received by the present HCD from the hub, then the present HCD unpacks the shared document from the system independent language and translates the document into a language preferred by the present HCD (step **634**). The HCD then processes the shared document according to the wishes of a user (step **636**) and then waits for the next event (step **601**).

If the event is not a shared document received from the hub, then the HCD determines if the event is a request by another HCD for information from the present HCD (step **638**). If the event is a request from another HCD, then the present HCD packs the answer in a system independent language (step **640**) and sends the answer to the hub (step **642**). The request may be for a list of shared documents present on the present HCD or for one of the shared documents present on the HCD. The HCD then waits for the next event (step **601**).

If the event is not a request by another HCD, then the HCD determines if the event is an end shared document session event (step **644**). If the event is an end shared document session event, then the session the HCDs participation in the session is terminated (step **646**). Otherwise, the HCD waits for the next event (step **601**).

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, and CD-ROMs and transmission-type media such as digital and analog communications links.

The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.